

*Amendments To The Claims:*

This listing of claims will replace all prior versions and listings of claims in this application:

*Listing of the Claims*

Claim 1 (previously presented): A method of data transfer in a virtual server environment of a computing network, the method comprising:

receiving a plurality of packets to be routed to or from a plurality of virtual servers operating in a single physical device;

providing an internal routing table for data link layer routing to or from selected ones of the virtual servers, wherein entries in the internal routing table are learned dynamically while processing selected ones of the received packets at a network layer; and

using the internal routing table for routing other ones of the received packets to or from the selected ones of the virtual servers at the data link layer,

wherein:

the selected ones of the received packets comprise, for each supported pair of input data link layer component and output data link layer component, a first-processed one of the packets which arrives using the input data link layer component and which is addressed to the output data link layer component; and

the other ones of the received packets comprise, for each of the supported pairs of input data link layer component and output data link layer component, subsequently-processed ones of the packets which arrive using the input data link layer component and which are addressed to the output data link layer component.

Claim 2 (currently amended): A method of data transfer in a virtual server environment of a ~~communications~~ computing network, the method comprising:

providing a concentrator that combines traffic from a plurality of virtual servers operating in a single physical device into a single outbound stream; and

routing packets of the combined traffic, further comprising steps of:

- intercepting packets of the traffic at a data link layer of a communications protocol stack;
- comparing a destination address of each intercepted packet to entries in a data link layer routing table comprising at least one entry, each entry specifying an input data link layer component, output data link layer component pair, to determine if a matching entry is present in the table, the matching entry specifying a data link layer component on which the intercepted packet arrived as the input data link layer component of the pair and the destination address of the intercepted packet as the output data link layer component of the pair;
- forwarding the intercepted packet to a higher layer of the communications protocol stack if the matching entry is not found, for routing by the higher layer; and
- performing data link layer routing of the intercepted packet, without intervention of the higher layer, if the matching entry is found.

Claim 3 (previously presented): The method according to Claim 2, wherein performing data link layer routing further comprises:

- replacing an inbound packet header of the intercepted packet with an outbound packet header using information from the matching entry, thereby creating a modified packet header; and
- forwarding the intercepted packet using the modified packet header.

Claim 4 (previously presented): The method according to Claim 2, wherein the entries in the data link layer routing table are dynamically learned and further comprising:

- adding a new entry to the table for each of the intercepted packets for which the matching entry is not found and for which the data link layer component on which the intercepted packet arrived and the output data link layer component that matches the destination address of the intercepted packet are both supported, the new entry specifying the data link layer component on which the intercepted packet arrived as the input data link layer

component of the pair and the output data link layer component that matches the destination address of the intercepted packet as the output data link layer component of the pair.

Claim 5 (previously presented): The method according to Claim 2, wherein at least one of the virtual servers is an application server.

Claim 6 (previously presented): The method according to Claim 2, wherein the virtual servers each operate in a logical partition within the single physical device.

Claim 7 (previously presented): The method according to Claim 2, further comprising deleting selected entries from the data link layer routing table when the selected entries become obsolete.

Claim 8 (currently amended): A system for data transfer in a virtual server environment of a ~~communications~~ computing network, comprising:

- means for providing a concentrator that combines traffic from a plurality of virtual servers operating in a single physical device into a single outbound stream; and

- means for routing packets of the combined traffic, further comprising:

- means for intercepting packets of the traffic at a data link layer of a communications protocol stack;

- means for comparing a destination address of each intercepted packet to entries in a data link layer routing table comprising at least one entry, each entry specifying an input data link layer component, output data link layer component pair, to determine if a matching entry is present in the table, the matching entry specifying a data link layer component on which the intercepted packet arrived as the input data link layer component of the pair and the destination address of the intercepted packet as the output data link layer component of the pair;

- means for forwarding the intercepted packet to a higher layer of the communications protocol stack if the matching entry is not found by the means for

comparing, for routing by the higher layer; and

means for performing data link layer routing of the intercepted packet, without intervention of the higher layer, if the matching entry is found by the means for comparing.

Claim 9 (previously presented): The system according to Claim 8, wherein the means for performing data link layer routing further comprises:

means for replacing an inbound packet header of the intercepted packet with an outbound packet header using information from the matching entry, thereby creating a modified packet header; and

means for forwarding the intercepted packet using the modified packet header.

Claim 10 (previously presented): The system according to Claim 8, wherein the entries in the data link layer routing table are dynamically learned, the dynamic learning further comprising:

means for adding a new entry to the table for each of the intercepted packets for which the means for comparing does not find the matching entry and for which the data link layer component on which the intercepted packet arrived and the output data link layer component that matches the destination address of the intercepted packet are both supported, the new entry specifying the data link layer component on which the intercepted packet arrived as the input data link layer component of the pair and the output data link layer component that matches the destination address of the intercepted packet as the output data link layer component of the pair.

Claim 11 (previously presented): The system according to Claim 8, wherein at least one of the virtual servers is an application server.

Claim 12 (currently amended): A computer program product for data transfer in a virtual server environment of a ~~communications~~ computing network, the computer program product embodied on computer-usable media having computer usable code embodied therewith, the computer program product comprising:

- computer readable program code for providing a concentrator that combines traffic from a plurality of virtual servers operating in a single physical device into a single outbound stream; and

- computer readable program code for routing packets of the combined traffic, further comprising:

- computer readable program code for intercepting packets of the traffic at a data link layer of a communications protocol stack;

- computer readable program code for comparing a destination address of each intercepted packet to entries in a data link layer routing table comprising at least one entry, each entry specifying an input data link layer component, output data link layer component pair, to determine if a matching entry is present in the table, the matching entry specifying a data link layer component on which the intercepted packet arrived as the input data link layer component of the pair and the destination address of the intercepted packet as the output data link layer component of the pair;

- computer readable program code for forwarding the intercepted packet to a higher layer of the communications protocol stack if the matching entry is not found by the computer readable program code for comparing, for routing by the higher layer; and

- computer readable program code for performing data link layer routing of the intercepted packet, without intervention of the higher layer, if the matching entry is found by the computer readable program code for comparing.

Claim 13 (previously presented): The computer program product according to Claim 12, wherein the computer readable program code for performing data link layer routing further

comprises:

computer readable program code for replacing an inbound packet header of the intercepted packet with an outbound packet header using information from the matching entry, thereby creating a modified packet header; and

computer readable program code for forwarding the intercepted packet using the modified packet header.

Claim 14 (previously presented): The computer program product according to Claim 12, wherein the entries in the data link layer routing table are dynamically learned, the dynamic learning further comprising:

computer readable program code for adding a new entry to the table for each of the intercepted packets for which the computer readable program code for comparing does not find the matching entry and for which the data link layer component on which the intercepted packet arrived and the output data link layer component that matches the destination address of the intercepted packet are both supported, the new entry specifying the data link layer component on which the intercepted packet arrived as the input data link layer component of the pair and the output data link layer component that matches the destination address of the intercepted packet as the output data link layer component of the pair.

Claim 15 (previously presented): The computer program product according to Claim 12, wherein at least one of the virtual servers is an application server.